INCREMENTAL INTERVAL ASSIGNMENT FOR MESH SCALING

Scott A. Mitchell

Center for Computing Research, Sandia National Laboratories, samitch@sandia.gov

ABSTRACT

Interval Assignment (IA) is the problem of selecting the number of mesh edges (intervals) for each curve. It ensures that surface quad meshes meet edge-to-edge on shared curve boundaries. To solve this, Incremental Interval Assignment (IIA) iteratively selects a set of coupled curves and increases each of their intervals by one. Compared to prior approaches, IIA is closest to network flows, and contrasts with optimization over integer variables. For mesh convergence studies, one starts with a hex mesh and needs to generate a series of finer meshes with essentially the same structure. The mesh should be refined at least a little bit everywhere. Mesh scaling achieves this by decomposing the input mesh into structured blocks, assigning new intervals to the three sides of each block, and remeshing blocks. (A variation remeshes highly unstructured regions with sweeping.) Mesh scaling allows the number of elements to grow more slowly than element refinement, which increases by at least 8×.

We describes IIA for the limited context of mesh scaling. Since we have a block decomposition, the curves are unambiguously partitioned into sets constrained to have equal intervals. We start with the original intervals. Using a priority queue, we select one curve (one set of curves), and increase its intervals by one, then put it back in the queue with a new priority. We terminate when no improvement is possible.

The advantages of IIA are both speed and output quality. Previously, mesh scaling IA used linear programming with branch and bound, whose runtime is potentially exponential in the number of curves. For some models it took days(!) and even failed to find any solution. IIA always works and runtime is linear in output intervals. In practice runtime is insignificant, less than a second, even for meshes of millions of elements. IIA comes closer to the target number of elements, the desired scaling factor for each curve, and geometrically distributing the change. This is achieved by running the priority queue several times with different priorities; some decrement intervals. (In contrast, prior IA approaches are typically tied to a single objective function.)

IIA works as follows. Original curves are called *simple*. To ensure at least a little bit of refinement everywhere, we group some adjacent simple curves into *aggregates* and put them on the queue as additional "curves." Aggregate curves, and simple curves not in any aggregate, are *top* curves. We use a series of six priority functions, where actually each function is a lexicographic ordering of up to four priorities to break ties. The first four queues ensure at least a little bit of increment happens everywhere. For the first queue, the user may (optionally) select a minimum increase for top curves. The second queue prioritizes simple curves that are in unchanged (or little-changed) top-curves. The third queue undoes unneeded increments: for a simple curve, can we decrement its intervals while still meeting the minimum increment? The fourth queue is run over top curves, and ensures every top curve is incremented at least once before any top curve is incremented twice. The secondary criteria is for each curve to individually meet the scaling factor. This is very effective at spreading out interval changes. The fifth and sixth queues focus on reaching the desired number of elements globally. The fifth queue runs over simple curves, prioritized by the increase in the number of hex elements, but keeping the total number of elements below about 1.01 times the target number of elements. The sixth queue tries to decrement curves as long as the total element count stays above the target.

Keywords: mesh refinement, hexahedral

This work was partially supported by the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.